



[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide



[Feedback](#) [Report a problem](#) [Satisfac](#)

Terms used

associative cache priority CAM BAM SRAM DRAM nonoverlapping overlapping overlapped nonoverlapped

Sort results by
Display results

[Save results to a Binder](#)

[Search Tips](#)

☐ Open results in a new window

[Try an Advanced Search](#)

[Try this search in The AC](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Re

1 [Scalable high-speed prefix matching](#)

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner

November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Full text available: [pdf\(933.02 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index te](#)

Finding the longest matching prefix from a database of keywords is an old problem with a number ranging from dictionary searches to advanced memory management to computational geometry. today's most frequent best matching prefix lookups occur in the Internet, when forwarding packet router. Internet traffic volume and link speeds are rapidly increasing; at the same time, a growing is increasing the size of routing tables against which p ...

Keywords: collision resolution, forwarding lookups, high-speed networking

2 [Data and memory optimization techniques for embedded systems](#)

P. R. Panda, F. Catthoor, N. D. Dutt, K. Danckaert, E. Brockmeyer, C. Kulkarni, A. Vandercappelle, P

April 2001 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volum

Full text available: [pdf\(339.91 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index te](#)

We present a survey of the state-of-the-art techniques used in performing data and memory-rela in embedded systems. The optimizations are targeted directly or indirectly at the memory subsys one or more out of three important cost metrics: area, performance, and power dissipation of the implementation. We first examine architecture-independent optimizations in the form of code trar We next cover a broad spectrum of optimizati ...

Keywords: DRAM, SRAM, address generation, allocation, architecture exploration, code transfor cache, data optimization, high-level synthesis, memory architecture customization, memory powe register file, size estimation, survey

3 [Predictor-directed stream buffers](#)

Timothy Sherwood, Suleyman Sair, Brad Calder

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Micro**

Full text available: [pdf\(187.39 KB\)](#) [ps\(1.12](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

[MB\)](#) [Publisher Site](#)

4 Intraprogram dynamic voltage scaling: Bounding opportunities with analytic modeling

Fen Xie, Margaret Martonosi, Sharad Malik

September 2004 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 1 Issue 1

Full text available:  pdf(980.11 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Dynamic voltage scaling (DVS) has become an important dynamic power-management technique. DVS tunes the power-performance tradeoff to the needs of the application. The goal is to minimize consumption while meeting performance needs. Since CPU power consumption is strongly dependent on supply voltage, DVS exploits the ability to control the power consumption by varying a processor's supply voltage and clock frequency. However, because of the energy and time overhead associated with ...

Keywords: Analytical model, compiler, dynamic voltage scaling, low power, mixed-integer linear

5 Reducing memory latency via non-blocking and prefetching caches

Tien-Fu Chen, Jean-Loup Baer

September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on support for programming languages and operating systems**, Volume 27 Issue 9

Full text available:  pdf(1.36 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

6 Concurrency, latency, or system overhead: which has the largest impact on uniprocessor DCache performance?

Vinodh Cuppu, Bruce Jacob

May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture**, Volume 29 Issue 2

Full text available:  pdf(904.17 KB)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Given a fixed CPU architecture and a fixed DRAM timing specification, there is still a large design space for DRAM system organization. Parameters include the number of memory channels, the bandwidth of each channel, burst sizes, queue sizes and organizations, turnaround overhead, memory-controller page protocols, assigning request priorities and scheduling requests dynamically, etc. In this design space, we seek to find the variation in application execution times: for example, ...

7 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Computer research**

Full text available:  pdf(4.21 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process diagrams are often used to obtain a better understanding of the execution of the application. The visualization we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are complex and do not provide the user with the desired overview of the application. In our experience, we display repeated occurrences of non-trivial communication patterns ...

8 System-level power optimization: techniques and tools

Luca Benini, Giovanni de Micheli

April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2

Full text available:  pdf(385.22 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This tutorial surveys design methods for energy-efficient system-level design. We consider electronic systems consisting of a hardware platform and software layers. We consider the three major constituents of a system that consume energy, namely computation, communication, and storage units, and we review methods for optimizing their energy consumption.

their energy consumption. We also study models for analyzing the energy cost of software, and n energy-efficient software design and compilation. This survery ...

9 Design and Implementation of High-Performance Memory Systems for Future Packet Buffer

Jorge García, Jesús Corbal, Llorenç Cerdà, Mateo Valero

December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Micro**

Full text available:  pdf(348.55 KB)


Additional Information: [full citation](#), [abstract](#), [index terms](#)

In this paper we address the design of a future high-speedrouter that supports line rates as high . Gb/s),around one hundred ports and several service classes. Buildingsuch a high-speed router wc technological problems,one of them being the packet buffer design, mainly becausein router desig to provide worst-case bandwidthguarantees and not just average-case optimizations.A previous p design provides worst-case bandwidthguarantees by using ...

10 High-speed policy-based packet forwarding using efficient multi-dimensional range matching

T. V. Lakshman, D. Stiliadis

October 1998 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIG conference on Applications, technologies, architectures, and protocols for com communication**, Volume 28 Issue 4

Full text available:  pdf(1.82 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index te](#)

The ability to provide differentiated services to users with widely varying requirements is becomin important, and Internet Service Providers would like to provide these differentiated services using shared network infrastructure. The key mechanism, that enables differentiation in a connectionles packet classification function that parses the headers of the packets, and after determining their c them based on administrative policies or re ...

11 Smart Memories: a modular reconfigurable architecture

Ken Mai, Tim Paaske, Nuwan Jayasena, Ron Ho, William J. Dally, Mark Horowitz

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual ir symposium on Computer architecture**, Volume 28 Issue 2

Full text available:  pdf(80.16 KB)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index te](#)

Trends in VLSI technology scaling demand that future computing devices be narrowly focused to i performance and high efficiency, yet also target the high volumes and low costs of widely applical purpose designs. To address these conflicting requirements, we propose a modular reconfigurable called Smart Memories, targeted at computing needs in the 0.1&mgr; technology generation. A S chip is made up of many processing tiles, each containing local ...

12 CRUSADE: hardware/software co-synthesis of dynamically reconfigurable heterogeneous re distributed embedded systems

Bharat P. Dave

January 1999 **Proceedings of the conference on Design, automation and test in Europe**

Full text available:  pdf(59.35 KB)

Additional Information: [full citation](#), [citations](#), [index terms](#)

13 A permutation-based page interleaving scheme to reduce row-buffer conflicts and exploit da

Zhao Zhang, Zhichun Zhu, Xiaodong Zhang

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Micro**

Full text available:  pdf(153.06 KB)  ps

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

(856.21 KB)  [Publisher Site](#)

14 External memory algorithms and data structures: dealing with massive data

Jeffrey Scott Vitter

June 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 2

Full text available:  pdf(828.46 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index to](#)


Data sets in large applications are often too massive to fit completely inside the computers internal resulting input/output communication (or I/O) between fast internal memory and slower external as disks) can be a major performance bottleneck. In this article we survey the state of the art in analysis of external memory (or EM) algorithms and data structures, where the goal is to exploit I/O to reduce the I/O costs. We consider a varie ...

Keywords: B-tree, I/O, batched, block, disk, dynamic, extendible hashing, external memory, hierarchical memory, multidimensional access methods, multilevel memory, online, out-of-core, secondary storage

15 Mining block correlations to improve storage performance

Zhenmin Li, Zhifeng Chen, Yuanyuan Zhou

May 2005 **ACM Transactions on Storage (TOS)**, Volume 1 Issue 2

Full text available:  pdf(1.02 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Block correlations are common semantic patterns in storage systems. They can be exploited for improving the effectiveness of storage caching, prefetching, data layout, and disk scheduling. Unfortunately, information about block correlations is unavailable at the storage system level. Previous approaches for discovering block correlations in file systems do not scale well enough for discovering block correlations in storage systems. In this paper we propose two algorithms, *C-Miner* and ...

Keywords: Storage management, block correlations, file system management, mining methods

16 The Clipper processor: instruction set architecture and implementation

W. Hollingsworth, H. Sachs, A. J. Smith

February 1989 **Communications of the ACM**, Volume 32 Issue 2

Full text available:  pdf(4.67 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index to](#)

Intergraph's CLIPPER microprocessor is a high performance, three chip module that implements a set architecture designed for convenient programmability, broad functionality, and easy future expansion.

17 A fully associative software-managed cache design

Erik G. Hallnor, Steven K. Reinhardt

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture**, Volume 28 Issue 2

Full text available:  pdf(117.18 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index to](#)

As DRAM access latencies approach a thousand instruction-execution times and on-chip caches grow to megabytes, it is not clear that conventional cache structures continue to be appropriate. Two key techniques—associativity and software management—have been used successfully in the virtual-memory domain to deal with disk access latencies. Future systems will need to employ similar techniques to deal with DRAM latencies. This paper presents a practical, fully associative cache design.

18 Full papers: Tree bitmap: hardware/software IP lookups with incremental updates

Will Eatherton, George Varghese, Zubin Dittia

April 2004 **ACM SIGCOMM Computer Communication Review**, Volume 34 Issue 2

Full text available:  pdf(189.39 KB)

Additional Information: [full citation](#), [abstract](#), [references](#)

Even with the significant focus on IP address lookup in the published literature as well as focus on commercial semiconductor vendors, there is still a challenge for router architects to find solutions to the problem of incremental updates.

simultaneously meet 3 criteria: scaling in terms of lookup speeds as well as table sizes, the ability speed updates, and the ability to fit into the overall memory architecture of an Level 3 forwarding packet processor with low systems cost overhead. I ...

19 A general framework for prefetch scheduling in linked data structures and its application to n prefetching

Seungryul Choi, Nicholas Kohout, Sumit Pamnani, Dongkeun Kim, Donald Yeung

May 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 2

Full text available:  pdf(2.45 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Pointer-chasing applications tend to traverse composite data structures consisting of multiple index chains. While the traversal of any single pointer chain leads to the serialization of memory operations, traversal of independent pointer chains provides a source of memory parallelism. This article investigates exploiting such *interchain memory parallelism* for the purpose of memory latency tolerance, using a technique called *multi-chain prefetching*. Previous work ...

Keywords: Data prefetching, memory parallelism, pointer-chasing code

20 A general-purpose compression scheme for large collections

July 2002 **ACM Transactions on Information Systems (TOIS)**, Volume 20 Issue 3

Full text available:  pdf(260.29 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

Compression of large collections can lead to improvements in retrieval times by offsetting the CPU costs with the cost of seeking and retrieving data from disk. We propose a semistatic phrase-based compression scheme called xray that builds a model offline using sample training data extracted from a collection, and compresses the entire collection online in a single pass. The particular benefits of xray are that it is applicable to applications where individual records or documents must be accessed randomly.

Keywords: phrase-based compression, random access, sampling

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows Media Player  Real Player

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	28	(Alex near2 Henderson).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:10
L2	21	(Walter near2 Croft).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:10
L3	20	1 and 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:12
L4	14	boundary adj addressable adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:13
L5	5911	content adj addressable adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:13
L6	414043	BAM or CAM	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:13
L7	4500	associative adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:17
L8	3094	primary adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:14
L9	5305	secondary adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:15

L10	4459	search adj key	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:15
L11	191	overlap\$4 adj entr\$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:15
L12	0	nonoverlap\$4 adj entr\$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:17
L13	131	binary adj range	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:16
L14	4019	conversion adj module	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:16
L15	101887	DRAM	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:16
L16	51470	SRAM	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:16
L17	23008	highest adj priority	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:16
L18	7735	match\$4 near2 entry	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:16
L19	712	6 and 7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:16

L20	75	10 and 19	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:17
L21	9	non adj overlap\$4 adj entr\$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:18
L22	191	11 or 21	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:18
L23	1	22 and 20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:18
L24	1	13 and 23	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/19 13:18